LEARNING OBJECT DETECTION USING MULTIPLE NEURAL NETWORKS

Keywords: object detection, neural networks, multiple neural networks

Abstract: Multiple neural network systems have become popular techniques for tackling complex tasks, often giving improved performance compared to a single network. In this study we propose an innovative detection algorithm in image analysis using a multiple neural network approach where many neural networks are jointly used to solve the object detection problem. We use a group of networks configured with different parameters and features, then combines them in order to obtain new networks. The topology of the set of neural networks is statically configured as a tree where the root node produces in output the detection map. This work represents a preliminary study through which we want to move from detection to segmentation and recognition of objects of interest. We have compared our model with other detection algorithms using a standard dataset and the results are encouraging. The results highlight the advantages and problems that will guide the evolution of the proposed model.

1 Introduction

Object detection is an important task in computer vision, it is a critical part in many applications such as content based image retrieval, understanding of a scene, automatic annotations, etc. However it is still an open problem due to the heterogeneity of some classes of objects to be detect and the complexity of the background in some images. The detection of an object in a scene allows to identify the object type and its location. The result should be the most accurate, because the performance of all the algorithms that can be applied subsequently to the objects identified, depends on it.

Usually the object of interest in a digital image is detected finding the bounding box which surround the object. The goal and the strength of this work consist in the detection of the object in a cognitive manner, locating the object through the use of a segmentation process. It allows to find the object of interest and at the same time to select only the pixels that belong to it. Figure 1 shows an example of detection and recognition in which the object of interest is a T-shirt. Detection and segmentation of an object in a single step is certainly a more complex and difficult operation instead of detect an object and perform a segmentation in a second step. In this work we use only the detection phase of our algorithm in order to compare the results with other object detection techniques despite the algorithm presented performs also a segmentation of the object of interest.

In literature there are many object detection approaches which can be classified as *bag of words* model (Csurka et al., 2004; Fei-fei, 2005; Schmid, 2006), *parts and structure* models (Fischler and Elschlager, 1973; Fergus et al., 2003; Crandall et al., 2005), *discriminative methods* (Bouchard and Triggs, 2005; Zhu and Yuille, 2006) and *combined segmentation and recognition* methods (Leibe and Schiele, 2003; Todorovic and Ahuja, 2006). In order to represent an image using the *bag of words* model, an image can be treated as a document. Like a document also the “words” (or pat-

(a) detection (b) segmentation

Figura 1: Example of an object detection phase using a bounding box (a) and an object detection and segmentation process by selecting the pixels belonging to the object of interest (b).

ches) into the images should be defined. One of the drawbacks of this method is that it ignores the spatial relationships between the image's patches. *Parts and structure models* refer to a broad class of object detection algorithms, which are used in various parts of the image separately, to determine if an object of interest is available and where it is located. These models differ significantly from the bag of words models, which explicitly ignore the position of image features. *Discriminative methods* include a lot of training techniques such as Boosting (Torralba et al., 2004), SVM (Dalal and Triggs, 2005), and make use of many promising features such as Haar (Viola and Jones, 2001) and HoG (Dalal et al., 2006). In this context the images are partitioned into a set of overlapping windows and the detection and recognition prediction is formulated as a classification problem. For each window we have to decide if it contains the object of interest or not. Models that fall into the category of *combined segmentation and recognition* provide an object segmentation as the process showed in Figure 1(b).

In this paper, choosing to work through the use of neural networks, it is possible to classify our algorithm in the category of the discriminative methods which go in the direction of the algorithms that combine segmentation and recognition.

The model proposed in this study is inspired by the human visual perception system. In fact, analyzing how the visual system works, a neuron n of the visual cortex receives a bottom-up signal X from the retina (lower-level-input) and a signal M from an object-model-concept m (top-down priming signal). The neuron n is activated if both signals are strong enough. The visual perception uses many levels in the transition from the retina to object perception. By analogy, we propose a *Multi-net system* (Sharkey, 1999) based on a tree-structure where leaf nodes represent the bottom-up signal extracted from the image and they are aggregated into a new signal in order to be transferred to the next level. The intermediate levels nodes represent the knowledge of the previous experience thus contributing at the activation of neurons, going in the direction of the root node.

In this paper the prediction of a trained network can become an input feature of a second neural model. Starting from the root node of a tree of neural networks, the training process is propagated to the leaf nodes of the tree (networks that depend only by simple features). During the detection phase, each node gives its generated map to its parent node until the map produced by the root node. Each node reads its information through a sliding window that receives all the input features. Neural networks are trained on the same set of training images and the prediction of each of them is combined in order to improve their generalization attitude. The peculiarity of the proposed model is that each level of the tree is guided by the same rules: for each node the perception of small and particular features or the knowledge of more abstract and complex concepts is performed by the same mechanism described above, but using a different node configuration.

2 The Proposed Method

In this section we present our model called Multi-Net for Object Detection (MNOD). The model proposed in this study consists of a tree of single networks $C^n_{\mathbf{P}}(\mathbf{F})$, where n is the node identifier, $\mathbf{P}$ corresponds to all the parameters from which the network configuration depends and $\mathbf{F}$ represents the set of features extracted from the input image.

Each neural network C^n uses a particular set of parameters $\mathbf{P}$ such as training epochs, number of neurons in the input layer, number of neurons in the hidden layer, etc.. The set of features $\mathbf{F}$ can be directly extracted from the input images (such as information on the edges, color, etc...)

or they can be the result of other neural models $C^n_{\mathbf{P}}$ previously trained on the same training set. The Figure 2 represents an example of a generic Multi-net tree, in this work named MNOD.

Each node $C^n_{\mathbf{P}}$ produces in output a map where the pixels containing higher values identify the objects of interest. As said in (Sharkey, 1999) it is possible to diversify a node $C^n_{\mathbf{P}}$ changing network initialization conditions, topologies, training algorithms and training data. Here, we change only the network topology considering the set of parameters $\mathbf{P}$. The main parameters associated which each node n are $\mathbf{P}_n = \{I_S, W_S\}$, where I_S represents the re-sized image dimension and W_S is the sliding window dimension. The structure of a single node used in this work, is showed in Figure 3. It consist in a Multi-Layer Perceptron (MLP) which receives the input values from a sliding window for each feature configured. We create an image for each feature and all these images are re-sized to the pre-definite dimension. By using different combinations of these two parameters it is possible to construct models specialized in the recognition of a specific element of interest in a particular scale. This concept is typical of Multi-net systems which use a "modular combination" of the neural networks (Sharkey, 1999). In this case the main problem is divided into subtasks and then it is possible to obtain a solution by combining different modules, or neural networks.

During the training phase all the input and output images of a node n are re-sized to the same size $I_S \in \mathbf{P}_n$ and from all the gray values in the sliding window we construct the input and desired output (as showed in Figure 3).

A single node in the proposed model can be trained or used directly in generalization unless it receives in input the output of another neural model (see for example the nodes $C^1_{\mathbf{P}_1}$ e $C^3_{\mathbf{P}_3}$ in the Figure 2). Otherwise (see, for example nodes $C^2_{\mathbf{P}_2}$ e $C^4_{\mathbf{P}_4}$ in the Figure 2) we first need to train and then to use the networks of child nodes. For more details on the algorithms used for training the model see the Algorithm 1 and the Figure 3, while for generalization see Algorithm 2 and Figure 3.

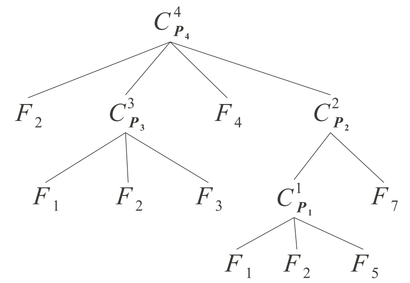

Figura 2: Generic structure of the proposed MNOD model. The nodes $C^n_{\mathbf{P}}$ represent the supervised neural models which receive their input directly from the input features, the nodes F_i are the feature maps extracted from the input images.

Algorithm 1 Creating the training set for a node $C^n_{\mathbf{P}}(\mathbf{F})$

Require: Let $D = \{(I^{in}_1, I^{out}_1), \ldots, (I^{in}_T, I^{out}_T)\}$ the set of images pairs;
for all $(I^{in}_t, I^{out}_t) \in D$ **do**
  extract $\mathbf{F} = \{F_1, \ldots, F_N\}$ from I^{in}_1 as input images for the node C^n
  if $\exists F_i \in \mathbf{F}$ which is the output of a child node C^m **then**
    execute this algorithm in order to create the training set for the node C^m
    train the node C^m
    use the trained model to create the image F_i
  end if
  Re size the features $F_1, \ldots, F_N$ and the output image at the dimension $I_S \in \mathbf{P}$
  Create an input and an output pattern for each position of the sliding window $W_S \in \mathbf{P}$ (step s depends by the image size I_S)
end for

---

3 Experiments

In this study, three types of experiments were conducted to analyze different aspects of the proposed model. We first analyzed the robustness of a single node (or single neural network) varying the size of the training set. The dataset is very simple and contains only images of cars positioned in the central part of the picture, with the same scale and same point of view. As a second test we used a more complex dataset, to study the main parameters of the MNOD models and its generalization attitude. Finally, we tried to compare our model with the results obtained by other methods using a standard dataset.

The two variables of interest to measure the accuracy of an object detection system, are the number of correct detection that we want to maximize, and the number of false detection that we want to minimize. When an object detection

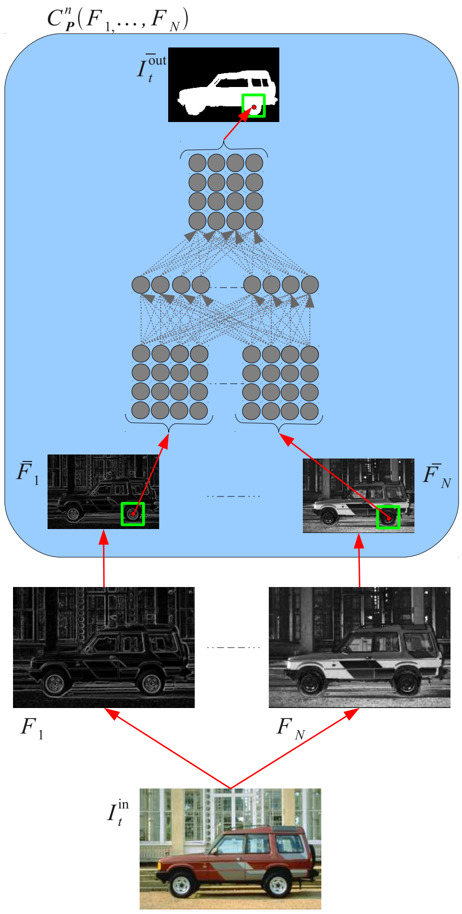

Figura 3: Generic structure of a node $C^n_{\mathbf{P}}(F_1, \dots, F_N)$. The features $F_1, \dots, F_N$ are created from each input images I_t^{in}. All the features images are re-sized in according to the parameter $I_S \in \mathbf{P}$ in $\bar{F}_1, \dots, \bar{F}_N$. The output image I_t^{out} is hand-segmented and contains higher values for the pixels belonging to the objects of interest. The input patterns of the neural model are generated using a sliding window of size $W_S \in \mathbf{P}$ which reads from the re sized images $\bar{F}_1, \dots, \bar{F}_N$, while the desired output is derived from the re sized output image $\bar{I}_t^{out}$.

system is really used, we are interested to know how many objects have been identified, and how often the items found are false. This compromise can be captured by studying the Precision-Recall curve (or PR-curves), where

$$P = \frac{correct}{actual}, \quad R = \frac{correct}{possible} \tag{1}$$

stating that *correct* is the number of objects correctly detected by the system, *actual* is the total number of objects recognized by the system, *possible* is the total number of objects we expected from system. A Precision equals to 1.0 means

Algorithm 2 Generalization of a single node $C^n_{\mathbf{P}}(\mathbf{F})$

Require: an image I^{in} to pass in input to the trained model

create $\mathbf{F} = \{F_1, \dots, F_N\}$ from I^{in} for the node C^n
if $\exists F_i \in \mathbf{F}$ such that F_i is the output of a child node C^m **then**
  apply this algorithm to the node C^m to generate each images F_i
end if
Re-size all the features $F_1, \dots, F_N$ at dimension $I_S \in \mathbf{P}$
Create an input pattern for each position of the sliding window $W_S \in \mathbf{P}$ (step 1)
Store output predictions of the trained node $C^n_{\mathbf{P}}(\mathbf{F})$
Compose the output image using the average activation values for each pixel

that each detected object is correct, but tells us nothing about the objects that have not been found. A high Precision value ensures that there are few false positives. A Recall measure equals to 1.0 tells us that each object was correctly identified, but tells us nothing about how many other items were incorrectly matched. A measure that usually is used to summarize Precision and Recall values is the F-measure $F = 1/(\lambda P + (1-\lambda)R)$ where usually $\lambda = 0.5$. Average Precision (AP) is another measure that approximates the area under the PR-curve and that, for its readability, is usually used to compare different object detection algorithms. For more details on these measures and how an object is considered correctly detected see (Everingham et al., 2005).

3.1 Evaluation of the basic idea

This first experiment shows the potentialities of a single node of the MNOD model. We used a simple dataset, the *ETHZ sideviews cars*[1], to study the generalization capability of a single node when the size of the training set varies.

The dataset contains only 50 images of cars. All the images are scaled so that a car object occupies approximately the same area (about 200 pixels wide). For each image, a hand-drawn figure-ground segmentation map (GT-Mask) is provided. The training set was built with only

[1] `http://www.vision.ee.ethz.ch/~bleibe/data/datasets.html#cars-side`

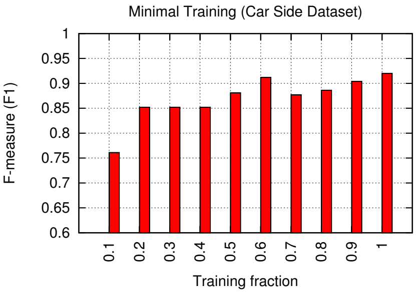

Figura 4: Average performance of the F1 measure varying the percentage of patterns belonging to the training set *Car Side*. The F1 measure was obtained using the same test set, performing 10 training/testing runs for each portion of the training that ranges from 10% to 100%.

26 images that are associated with a GT-Mask image that identify the object (see Figure 5(b)). In this way we obtained a training set consisting of 26 images and a test set of 24 images.

In this experiment, the neural network reads in input the intensity values of pixels contained in a 8×8 window, which also corresponds to the size of the scaled image (see Figure 3). In this way, the MLP network has 64 neurons on each of the three layers: input, hidden and output. The network was trained for 300 epochs to obtain the following result: Precision= 0.92 (22/24), Recall=0.92 (22/24), F-Measure=0.92. In Figure 5 an example of image detection, using the trained neural model on a test image.

The plot in Figure 4 shows the measure $F1$ computed on the test set, depending on the number of examples of the training set that, for this experiment, is equals to the number of images. More specifically, using the original partitions training-test, we trained the network by varying the percentage of training images and extract them randomly at each step. The considered percentages range from 10% to 100% with a step equal to 10%. For each percentage, the net was trained ten times, randomly changing the condition of the network initialization. Selecting at minimum 20% of pictures belonging to the training set, we get very similar results to those obtained with the entire training set. This means that, for this type of data, in order to train a neural model able to detect an object belonging to the class of interest, we need about 10 GT-Mask images.

From this first experiment, we may conclude that a single neural network is sufficient to recognize a class of objects seen at the same scale and from the same point of view. Things get complicated when the images contain the object of interest seen at different scales and from all the possible viewpoints. In this case we can not use only an input window for the neural network equal to the size of the image, because scaling the image we may deform the object or risk to lose content information. For this reason, in the following section, we try to better analyze the influence of MNOD parameters, using a heterogeneous dataset.

(a) input (b) mask (c) detection

Figura 5: An example of test image belonging to the dataset *ETHZ sideviews cars* included in *VOC2005* (a), the GT-Mask image that identifies the car (b) and the 8×8 neural network output enlarged to original size (c). In (c) are also visible two rectangles: the red one indicates the detection made by our model while the green one is the truth.

3.2 Analysis of key parameters

In this experiment we analyzed the behavior of the MNOD model varying its key parameters. In particular, we analyzed the influence of the image size I_S and the sliding window size W_S parameters, applied to nodes (or networks) configured with different features.

The dataset used here is more complex than the previous and contains real images of cars viewed from different viewpoint: side, rear, top, front, partial (see some examples in Figure 7). In particular, we used the dataset called *TU-Graz cars* contained in *VOC2005* (Everingham et al., 2005).

For this experiment we used a limited set of features that highlight some information on color and high frequencies available in the images. In particular, we used the features *Saturation* (Sa), *Hue* (Hu), *Sobel* (So) e *Horizzontal Edges* (EH).

The MNOD topology is shown in figure 6 and was fixed before starting the training phase. Nodes were configured randomly by tying the window size $W_S < I_S$ and trying some combinations of parameters (W_S, I_S) leading to an increase in detection accuracy. The size of the sliding window ranges from 1×1 to 9×9, while the images were scaled proportionally so that the smal-

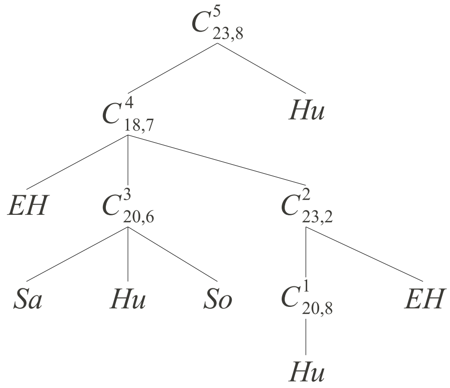

Figura 6: An configuration example where each node increases the accuracy measures (see Table 1 for details) using the output of child nodes.

ler side is in the range $[3, 24]$. The plots in Figure 8 show the behavior of the F-measure in the space (W_S, I_S). It is evident that the range $[F1_{min}, F1_{max}]$ and the activation area of the $F1$ measure increases, moving from leaf nodes towards the root node C^5.

Based on the results obtained with this first analysis we chosen a configuration for each node of the model shown in Figure 6. Each node C^n was trained with the following parameters: $C^3_{20,6}(Sa, Hu, So)$, $C^1_{20,8}(Hu)$, $C^2_{23,2}(EH, C^1)$, $C^4_{18,7}(C^3, C^2, EH)$ and $C^5_{23,8}(Hu, C^4)$.

From Table 1, we may observe that each node of the selected configuration, if it receives in input the output of one or more nodes, increases the generalization capability. In fact, the Table 1 shows an increase in P, R and F1, comparing a node with its child nodes. For example, the last row of the table shows the network $C^5_{23,8}(Hue, C^4)$ that takes advantage of the map generated by the network C^4.

Tabella 1: Results obtained using some different random configurations. $C_{x,y}(M_1, M_2, \dots)$ is a neural network that uses an image size $x \times x$, a sliding window size $y \times y$, reading the input values from all the feature M_i.

Configuration	P	R	F1
$C^3_{20,6}(Sa, Hu, So)$	0,11	0,11	0,11
$C^1_{20,8}(Hu)$	0,02	0,02	0,02
$C^2_{23,2}(EH, C^1)$	0,05	0,10	0,07
$C^4_{18,7}(C^3, C^2, EH)$	0,24	0,19	0,21
$C^5_{23,8}(Hu, C^4)$	0,33	0,26	0,29

All the nodes in Table 1 were trained using 108

Figura 7: Some examples of images belonging to the dataset *TU-Graz cars*.

training images. Results shown in table are taken from the test set consisting of 128 never seen images. The train and test sets are the same done in the *VOC2005 dataset 1*. Each network was trained using the learning algorithm Rprop (Resilient Backpropagation) proposed by Riedmiller and Braun (Riedmiller and Braun, 1993).

3.3 Comparisons

In (Everingham et al., 2005) several innovative methods for object detection were evaluated on a dataset containing four classes of objects: *motorbikes*, *bicycles*, *people* and *cars*. The training and test sets contain objects having substantial variations in terms of scale, occlusion, and variability within the class. These aspect of the dataset allow to better assess an algorithm and to highlight the benefits and drawbacks of the same algorithm. Another advantage of this dataset is the possibility to compare an algorithm with the whole community that used it.

Our algorithm requires the presence of a GT-Mask for each training image, and not all the four classes of objects have that information associated with each training image. For this reason we used only three classes of objects *bicycles*, *people* and *cars* and for each class we use only the training image having a GT-Mask.

The metrics used here are the same as that used in (Everingham et al., 2005): the PR-curve and the Average Precision (AP).

The network showed in Figure 9 represents the structure used in all the three classes of objects. The structure of the model used in this experiment was built with the following rule: we build some of the nodes which receive in input some features and are configured with a small I_S value, the maps generated by these nodes are passed in input to other nodes that increase the I_S value, and so on. In this way we can see a refinement of the segmentation result through every level of the tree towards the root node.

From this experiment we found that the main

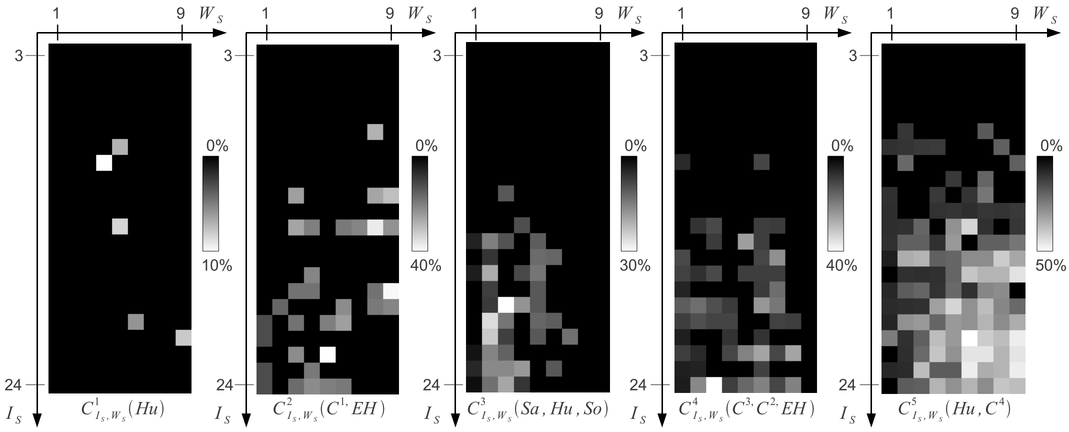

Figura 8: Each image shows how the F-measure varies in the parameter space (I_S, W_S) for each node $C^1_{I_S,W_S}$ - $C^5_{I_S,W_S}$ shown in Figure 6. A white pixel represents a F-measure equals to the maximum value. Each network was trained on 10 images of the training set and the F-measure was computed on the first 10 images of the testing set *TU-Graz cars.*

problem of our algorithm lies in the dataset with large differences of scale. Then, trying to force in input only the area containing the object (see some examples showed in the first row of the Figure 10), we note that the results increase significantly (see MNOD-SI in Table 2). The model so trained and tested was renamed MNOD-SI.

In this work a detailed analysis on the best features for this model has not been done. For this experiment we used a limited set of features that highlight some information on color and high frequencies present in the images. In particular, we used the features *Saturation* (Sa), *Hue* (Hu), *Brightness* (Br), *Horizzontal Edges* (EH), *Vertical Edges* (EV) and *Histogram of Gradient* (HoG) configured in different ways. The parameter s in EH_s and EV_s is the scale used on the input image before to compute the feature. The parameters (s, b) in $HoG_{s,b}$ are the scale used on the input image and the block size respectively. The HoG feature, as in (Dalal and Triggs, 2005), reads the input values from a sliding window of histograms (cell block). In this work each histogram is composed by four bins, and each cell block is an area of 6×6 pixels.

In Table 2 we compare our model with the method presented in (Dalal and Triggs, 2005), one of the best works of the competition (Everingham et al., 2005), while in Figure 11 the trend of the PR-curve using MNOD-SI is shown for the three classes of objects. The AP measure on the three classes of problems considered was computed using the same configuration showed in Figure 9 and the standard test set called *test1* for each class of objects of the *VOC 2005 Dataset 1.*

It is obvious that we could obtain better results by making a thorough search of the parameters and of the features, as evidenced in the study showed in Figure 8, but we will deal with this issue in a future work. Starting from the results obtained using the model MNOD-SI we are working for a new version able to exploiting the different scales in images of a particular domain. A very interesting result is the fact that the model has a similar behavior on different classes of objects.

Tabella 2: Average precision for the object detection problem applied to the test set called *test1* in *VOC 2005 Dataset 1.* The comparison results are taken from (Everingham et al., 2005).

Method	Bicycles	People	Cars
MNOD	0.01	0.02	0.06
MNOD-SI	0.2	0.21	0.42
Boosted-Histogram	037	0.25	0.663
TU-Darmstadt	-	-	0.489
Edinburg	0.19	0.002	0.00
INRIA-Dalal	-	0.01	0.61

4 Conclusions

In this paper we described an object detection algorithm based on a multiple networks system. It is composed by a set of neural networks aggregated together to provide a single output result. The model was tested and the results were pre-

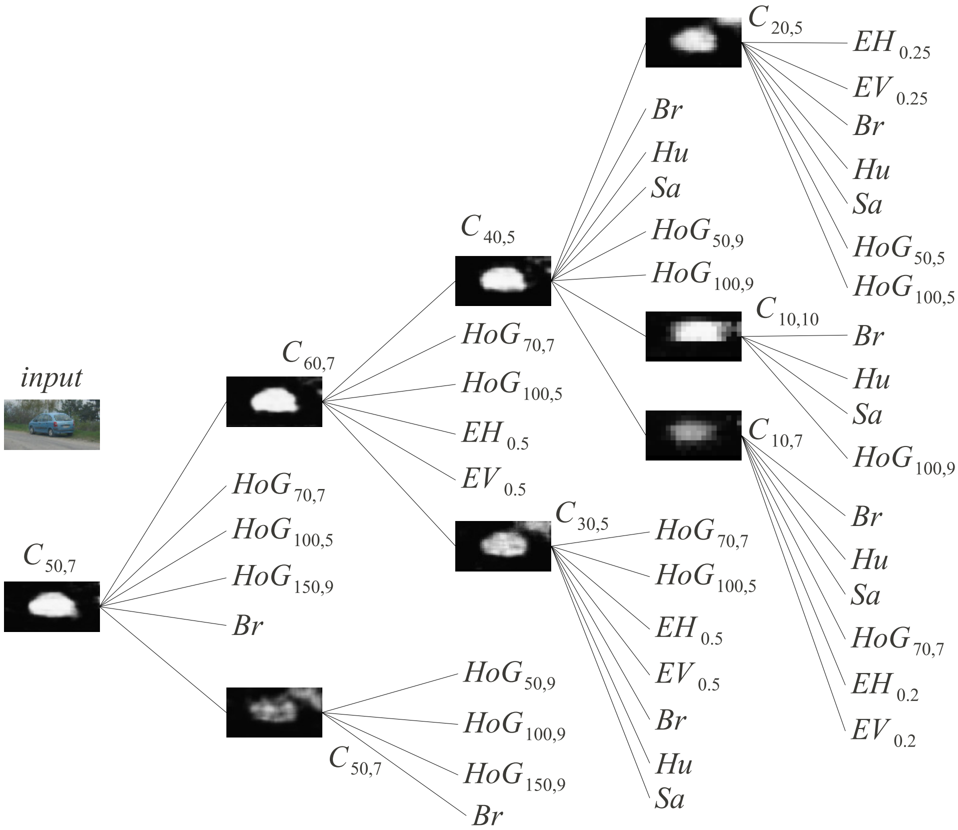

Figura 9: MNOD configuration used for the three categories of objects in the dataset VOC2005. The model displays the maps produced by each node $C_{\mathbf{P}}$ and the features used in input, all related to the *input* image of the testing set.

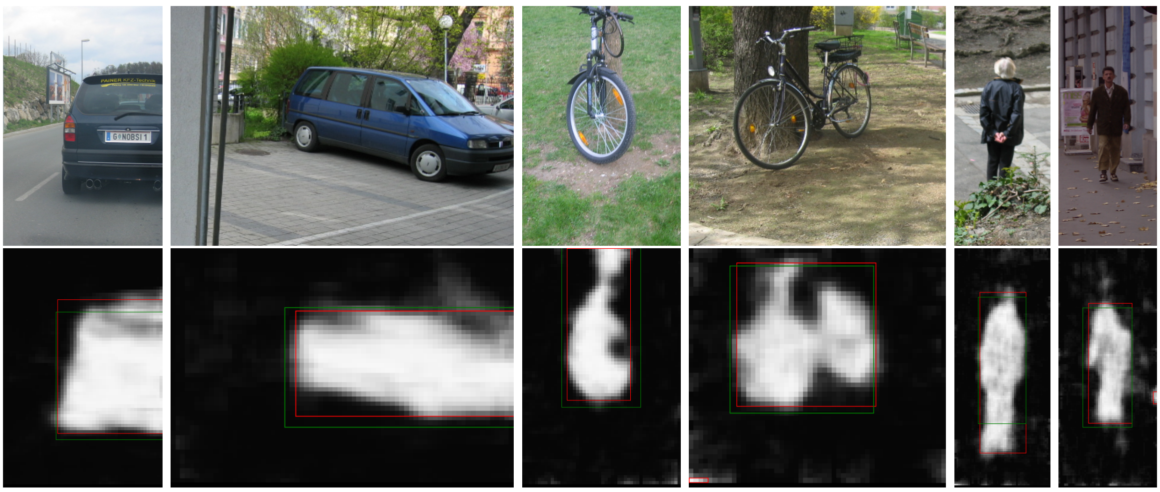

Figura 10: On the first line there are some image examples of the dataset VOC2005. On the second line the maps produced by the trained model and the desired bounding box in green.

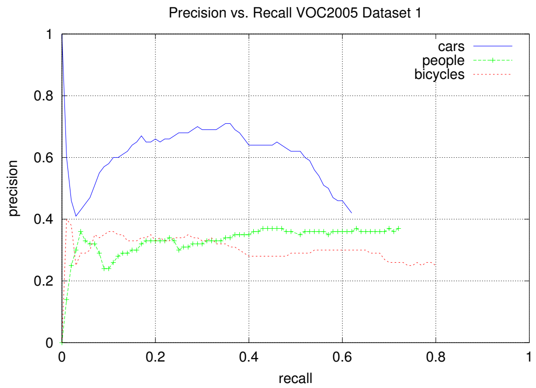

Figura 11: PR curve for the three classes of objects *bicycles*, *people* e *cars*. The model used for this graph is MNOD-SI.

sented, highlighting the potentialities of our solution. The proposed algorithm can be configured for different classes of object and its nodes may consist of different types of neural learning strategies. This type of algorithm, here mainly used for detection, provides an interesting segmentation result that we intend to exploit in future works.

We got good results on many standard dataset compared to other object detection algorithms. Moreover, the results show that our algorithm is robust to the change of perspective for the same object and at the same time, as it is robust for objects of the same type but different forms in different poses or even articulated and occluded.

The proposed algorithm suffers from some issues related to datasets with conspicuous differences of scale for the objects that we want to detect. Therefore we are working on a different strategy to overcome this problem. Detection results are dependent on network topology and from the selected set of features. Then, we are analyzing new search algorithms able to select the best network configuration.

REFERENCES

Bouchard, G. and Triggs, B. (2005). Hierarchical part-based visual object categorization. In Proc. CVPR, pages 710–715.

Crandall, D., Felzenszwalb, P., and Huttenlocher, D. (2005). Spatial priors for part-based recognition using statistical models. In Proc. CVPR, pages 10–17.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In In Workshop on Statistical Learning in Computer Vision, ECCV, pages 1–22.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In Proc. CVPR, pages 886–893.

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In Proc. European Conference on Computer Vision. Springer.

Everingham, M., Zisserman, A., Williams, C. K. I., Gool, L. V., Allan, M., Bishop, C. M., Chapelle, O., Dalal, N., Deselaers, T., Dork, G., Duffner, S., Eichhorn, J., Farquhar, J. D. R., Fritz, M., Garcia, C., Griffiths, T., Jurie, F., Keysers, D., Koskela, M., Laaksonen, J., Larlus, D., Leibe, B., Meng, H., Ney, H., Schiele, B., Schmid, C., Seemann, E., Shawe-taylor, J., Storkey, A., Szedmak, O., Triggs, B., Ulusoy, I., Viitaniemi, V., and Zhang, J. (2005). The 2005 pascal visual object classes challenge. In Selected Proceedings of the First PASCAL Challenges Workshop.

Fei-fei, L. (2005). A bayesian hierarchical model for learning natural scene categories. In Proc. CVPR, pages 524–531.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In Proc. CVPR, pages 264–271.

Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. Computers, IEEE Transactions on, C-22(1):67 – 92.

Leibe, B. and Schiele, B. (2003). Interleaved object categorization and segmentation. In Proc. BMVC, pages 759–768.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, pages 586–591.

Schmid, C. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proc. CVPR, pages 2169–2178.

Sharkey, A. J. (1999). Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, chapter Multi-Net Systems. Springer.

Todorovic, S. and Ahuja, N. (2006). Extracting subimages of an unknown category from a set of images. In Proc. CVPR, pages 927–934.

Torralba, A., Murphy, K. P., and Freeman, W. T. (2004). Sharing features: Efficient boosting procedures for multiclass object detection. In Proc. CVPR, pages 762–769.

Viola, P. and Jones, M. (2001). Robust real-time object detection.

Zhu, L. and Yuille, A. (2006). A hierarchical compositional system for rapid object detection. In Proc. NIPS.